

# 한국어 문장 부호를 조판하는 kopunct 패키지\*

강우현

[kangwh.2003@gmail.com](mailto:kangwh.2003@gmail.com)

2025년 1월 23일

## 요 약

kopunct 패키지는 L<sup>A</sup>T<sub>E</sub>X에서 한국어 문장 부호를 올바르게 조판하는 명령어를 제공한다. 이 패키지를 불러오면 한국어 키보드에 등장하지 않는 가운뎃점과 낫표, 화살괄호, 줄임표를 쉽게 입력할 수 있으며, 레거시 엔진에서는 더 보기 좋은 물결표를 얻을 수 있다.

## 차 례

<b>1 불러오기</b>	<b>2</b>
<b>2 기본 사용 방법</b>	<b>2</b>
2.1 가운뎃점과 줄임표 . . . . .	2
2.2 낫표와 화살괄호 . . . . .	3
<b>3 물결표</b>	<b>4</b>
3.1 물결표 세부 조정 . . . . .	4
<b>4 oblvioir 호환 모드</b>	<b>5</b>
<b>5 패키지 옵션</b>	<b>5</b>
5.1 재정의 방지 옵션 . . . . .	5
5.2 oblvioir 클래스 호환 옵션 . . . . .	6
<b>6 구현</b>	<b>6</b>
6.1 패키지 정보 . . . . .	6
6.2 옵션 선언 . . . . .	6
6.3 패키지 불러오기 . . . . .	7

---

\*이 문서는 2025년 1월 23일에 작성된 kopunct 패키지 버전 1.1 에 대응한다.

6.4 매크로 정의 . . . . .	7
6.4.1 가운데뿔점 . . . . .	7
6.4.2 줄임표 . . . . .	7
6.4.3 낫표와 화살괄호 . . . . .	8
6.4.4 물결표 재정의 . . . . .	9
6.5 oblivoir 호환 매크로 정의 . . . . .	10
6.5.1 가운데뿔점 . . . . .	10
6.5.2 줄임표 . . . . .	10
6.5.3 낫표와 화살괄호 . . . . .	11
6.5.4 줄표 . . . . .	11

## 1 불러오기

다른 패키지들과 마찬가지로 전처리부에

```
\usepackage{kopunct}
```

를 입력하여 패키지를 불러오면 된다.

유의 사항에는 다음과 같은 것들이 있다.

- $ko_{\text{TeX}}$ 이나 `polyglossia` 등 한국어 조판과 관련된 패키지를 불러온 후에 이를 불러오는 것이 좋다.
- 레거시 엔진에서는 물결표를 조정하는 데 `graphicx` 패키지를 사용하므로, 이 패키지를 불러와야 한다. `kopunct`는 `graphicx`를 자동으로 불러오지 않는다. 패키지를 불러오는 순서는 중요하지 않다.
- 레거시 엔진에서는 T1 인코딩을 사용해야 한다. 이 인코딩을 사용하지 않는 경우 가운데뿔점이 조판되지 않는다.

## 2 기본 사용 방법

이 패키지에서 제공하는 문장 부호 조판 명령어를 설명한다. 이번 절에서 소개하는 명령어들은 `oblivoironly` 옵션을 지정한 경우 사용할 수 없다.

### 2.1 가운데뿔점과 줄임표

`\cdot` 가운데뿔점은 `\cdot`으로, 3점 줄임표는 `\cdots`로 입력한다. 현재 모드에 상관없이 동 `\cdots` 일한 명령어로 동일한 모양의 기호를 조판할 수 있다. 6점 줄임표는 `\cdots`를 두 번 `\Cdots` 입력해도 되지만, 간편하게 `\Cdots`로도 입력할 수 있다.

### 예시

아침\cdot 점심\cdot 저녁 → 아침·점심·저녁

이\cdots 이럴 수가\Cdots. → 이…이럴 수가…….

`\hellipsis` 유니코드 엔진의 `ko.TeX`에서는 6점 줄임표를 조판하는 명령어로 `\hellipsis`를 제공하는데,<sup>1</sup> `kopunct`는 이 명령어가 정의되어 있지 않은 경우에 한하여 `\hellipsis`를 정의해 준다.

### 예시

이럴 수가\hellipsis. → 이럴 수가…….

`\dots` 한편, 유니코드 엔진에서 `\dots`와 `\ldots`는 레거시 엔진의 줄임표(...)와 다른 `\ldots` 모양의 줄임표를 조판한다. 이 패키지는 엔진에 관계없이 `\dots`와 `\ldots`가 동일한 모양의 줄임표(영어 등에서 사용하는 모양)를 조판하도록 한다.

### 예시

My brown fox is \dots gone! → My brown fox is ...gone!

## 2.2 낫표와 화살괄호

`\nat` 흘낫표는 `\nat`으로 입력한다. 인자를 입력하면 그 내용이 흘낫표에 싸인 채로 조판 `\Nat` 된다. ‘n’을 대문자 ‘N’으로 바꾸면 겹낫표가 조판된다.

### 예시

`\nat{흘낫표}` → 「흘낫표」

`\Nat{겹낫표}` → 『겹낫표』

`\hwa` 화살괄호는 `\hwa`와 `\Hwa`로 입력한다. 앞과 마찬가지로 소문자 `\hwa`는 홑화살괄 `\Hwa` 호, 대문자 `\Hwa`는 겹화살괄호를 조판한다.

### 예시

`\hwa{홑화살괄호}` → 〈홑화살괄호〉

`\Hwa{겹화살괄호}` → 《겹화살괄호》

`\lnat` 여는 쪽만 입력하려면 ‘l’을, 닫는 쪽만 입력하려면 ‘r’를 역슬래시 뒤에 입력하면 `\rnat` 된다. 이때에는 인자를 지정하지 않는다.

`\lNat`

### 예시

`\nNat`

`\lhwa` `\lnat` 흘낫표 `\rnat` → 「흘낫표」

`\rhwa` <sup>1</sup>어떤 이유에서인지 `LuaTeX-ko`나 `XgTeX-ko`의 설명서에는 이 사실이 나와 있지 않다.

`\lHwa`

`\nHwa`

$\backslash\text{Nat}$  겹낫표 $\backslash\text{rNat}$  → 『겹낫표』  
 $\backslash\text{lhwa}$  흘화살괄호 $\backslash\text{rhwa}$  → 〈흘화살괄호〉  
 $\backslash\text{Hwa}$  겹화살괄호 $\backslash\text{rHwa}$  → 《겹화살괄호》

### 3 물결표

(L<sup>A</sup>)T<sub>E</sub>X에서 물결표는  $\backslash\text{textasciitilde}$  또는  $\sim$ 로 조판한다. 유니코드 엔진에서는 우리가 생각하는 모양의 물결표가 그려지지만, 레거시 엔진에서는 악센트로 쓰이는 형태의 물결표가 그려진다. 비슷한 모양의 수학 기호를 조판하는  $\sim$ 을 대안으로 제시하는 이도 있지만, 이 기호를 사용하면 문장부호만 도드라져 보인다. 이는 ‘실제’ 물결표를 사용한 것도 아니므로 검색이나 복사 기능도 제대로 작동하지 않는다.

#### 예시

$2010\sim 2015$  → 2010~2015  
 $381\sim 403$  → 381~403

$\sim$  kopunct는 매크로를 재정의함으로써 레거시 엔진에서도 물결표가 바른 모양으로 그려지도록 한다. 패키지를 불러온 뒤  $\sim$ 을 입력하면 된다.  $\sim$ 의 틸데 악센트를 조판하는 기능도 정상적으로 작동한다. 만약 — 어떤 이상한 이유에서든 — 틸데 악센트를 단독으로 출력해야 한다면  $\backslash\text{textasciitilde}$ 를 입력하면 된다.<sup>2</sup>

#### 예시

$\text{제}1\sim\text{제}2\text{장}$  → 제1~2장  
 $\text{Espa}\sim\text{no}\text{l}$  → Español  
 틸데 악센트( $\backslash\text{textasciitilde}$ ) → 틸데 악센트( $\sim$ )

#### 3.1 물결표 세부 조정

kopunct 패키지는 T<sub>E</sub>X의 기본 로마자 글꼴인 Computer Modern Roman에 맞추어 틸데 악센트가 물결표처럼 보이도록 조정해 두었다. 따라서 기본 글꼴이 바뀐다면 물결표가 어울리지 않을 수 있다. 사용자는 내부 변수를 수정함으로써 현재 글꼴에 어울리도록 물결표의 크기와 위치를 조정할 수 있다.

$\backslash\text{TildeScaleFactor}$   $\backslash\text{TildeScaleFactor}$ 는 틸데를 확대할 배율로, 기본값은 1.5이다.  $\backslash\text{TildeRaise}$   $\backslash\text{TildeRaise}$ 는 틸데의 상하 이동 값으로, 기본값은  $-1.35\text{ex}$ 이다(음수이면 틸데가 아래로 내려간다).  $\backslash\text{TildeHeight}$ 는 물결표 문자의 높이 값으로, 기본값은  $1\text{ex}$ 이다. 이때  $\backslash\text{TildeRaise}$ 와  $\backslash\text{TildeHeight}$ 의 단위는 ex 또는 em으로 지정해야 글꼴의 크기 변화에 대응할 수 있다.

<sup>2</sup>OT1 인코딩을 사용하는 경우에는  $\backslash\text{textasciitilde}$ 를 사용하더라도 물결표가 출력된다. 이때에는  $\backslash\text{symbol}\{‘\sim’\}$ 를 입력하면 틸데 악센트를 얻을 수 있다.

표 1: kopunct 기본 명령어와 oblivoir 호환 명령어 대조표

문장 부호	kopunct 기본 명령어	oblivoir 호환 명령어
가운뎃점	<code>\cdot</code>	<code>\cntrdot</code>
3점 줄임표(한국어)	<code>\cdots</code>	<code>\cntrdots</code>
3점 줄임표(영어)	<code>\ldots</code> 또는 <code>\dots</code>	<code>\obldots</code>
6점 줄임표	<code>\Cdots</code> 또는 <code>\hellipsis</code>	<code>\obellipsis</code>
홀낫표	<code>\nat{...}</code> 또는 <code>\lnat ...\rnat</code>	<code>\snm{...}</code>
겹낫표	<code>\Nat{...}</code> 또는 <code>\lNat ...\rNat</code>	<code>\bnm{...}</code>
홀화살괄호	<code>\hwa{...}</code> 또는 <code>\lhwa ...\rhwa</code>	<code>\cnm{...}</code>
겹화살괄호	<code>\Hwa{...}</code> 또는 <code>\lHwa ...\rHwa</code>	<code>\ccnm{...}</code>

## 4 oblivoir 호환 모드

oblivoir 클래스는 한국어 문장 부호를 조판하는 명령어를 기본 제공한다. 패키지를 불러올 때 oblivoir 옵션을 지정하면 기본 명령어와 oblivoir 클래스의 명령어를 함께 사용할 수 있으며, oblivoironly 옵션을 지정하면 기본 명령어는 사용할 수 없고 oblivoir의 명령어만을 사용할 수 있다. 두 종류의 명령어의 대조표는 표 1에서 확인할 수 있다.

## 5 패키지 옵션

현재 버전의 kopunct는 `nocdot`, `nocdots`, `notilde`, `oblivoir`, `oblivoironly` 이렇게 다섯 개의 옵션을 제공한다.

### 5.1 재정의 방지 옵션

kopunct는 `\cdot`, `\cdots`, `\textasciitilde`를 재정의함으로써 문장 부호를 조판하는 명령어를 제공한다. 이 과정에서 예기치 못한 문제가 생길 수 있기 때문에, kopunct는 이들을 재정의하지 않도록 지시하는 옵션을 제공한다.

`nocdot`은 `\cdot`을 재정의하지 않도록 한다. 따라서 가운뎃점은 소스 코드에 직접 입력하거나 `\textperiodcentered`를 사용하여 입력해야 한다.

`nocdots`는 `\cdots`를 재정의하지 않도록 한다. 레거시 엔진에서는 `\unichar{"2026}`을 입력하여, 유니코드 엔진에서는 `\textellipsis`나 `\dots`, `\ldots`를 사용하여 3점 줄임표를 입력할 수 있다. 유니코드 엔진에서는 줄임표 문자를 소스 파일에 직접 입력해도 된다.

`notilde`는 레거시 엔진에서 `\~`를 재정의하지 않도록 한다. 유니코드 엔진에서는 아무런 효과가 없다.

## 5.2 oblivoir 클래스 호환 옵션

oblivoir 클래스는 한국어 문장 부호를 조판하는 자체적인 매크로를 제공한다. kopunct는 이 클래스와의 호환 기능을 제공한다.

oblivoir 옵션은 kopunct의 기본 명령어와 함께 oblivoir의 명령어를 사용할 수 있도록 한다. 반면, oblivoironly 옵션은 kopunct의 기본 명령어를 정의하지 않고 oblivoir의 문장 부호 명령어만을 정의한다. 호환 명령어는 본래의 클래스와 일부 다르게 동작할 수 있다.

oblivoironly를 지정한 경우, \cdot, \cdots, \~가 모두 재정의되지 않으므로 nocdot, nocdots, notilde는 아무 효과를 내지 않는다.

## 6 구현

### 6.1 패키지 정보

패키지의 정보를 제공한다.

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{kopunct}[2025/01/20 v1.2
3   Package for providing macros to input Korean punctuation marks]
```

### 6.2 옵션 선언

먼저 옵션을 처리하기 위한 스위치를 정의한다. 정의한 각 스위치는 false로 초기화한다.

```
4 \newif\ifkopunct@nocdot\kopunct@nocdotfalse
5 \newif\ifkopunct@nocdots\kopunct@nocdotsfalse
6 \newif\ifkopunct@notilde\kopunct@notildefalse
7 \newif\ifkopunct@oblivoir\kopunct@oblivoirfalse
8 \newif\ifkopunct@oblivoironly\kopunct@oblivoironlyfalse
```

이후 옵션을 선언한다.

```
9 \DeclareOption{nocdot}{\kopunct@nocdottrue}
10 \DeclareOption{nocdots}{\kopunct@nocdotstrue}
11 \DeclareOption{notilde}{\kopunct@notildetrue}
12 \DeclareOption{oblivoir}{\kopunct@oblivoirtrue}
13 \DeclareOption{oblivoironly}{%
14   \kopunct@oblivoirtrue
15   \kopunct@oblivoironlytrue
16 }
```

옵션 선언이 완료되었으면 옵션을 처리한다.

```
17 \ProcessOptions\relax
```

## 6.3 패키지 불러오기

레거시 엔진과 유니코드 엔진을 구분하기 위해 `iftex` 패키지를 불러온다.

```
18 \RequirePackage{iftex}
```

이 패키지는 줄임표를 정의할 때와 물결표를 재정의할 때 사용된다.

## 6.4 매크로 정의

기본 매크로는 `\ifkopunct@oblivioironly` 스위치가 `false`일 때에만 정의된다.

```
19 \ifkopunct@oblivioironly\else
```

### 6.4.1 가운데점

`\cdot` 텍스트 모드에서는 가운데점, 수식 모드에서는 내적 기호를 조판하도록 `\cdot`을 재정의한다. 이 작업은 `\ifkopunct@nocdot` 스위치가 `false`일 때에만 진행된다.

```
20 \ifkopunct@nocdot\else
```

`\math@cdot` `\cdot`의 정의에는 `\math@cdot`과 `\text@cdot`이 필요하다. `\math@cdot`은 수식 모드에서 내적 기호를 조판할 때 실제로 사용할 제어 문자열, `\text@cdot`은 텍스트 모드에서 가운데점을 조판할 때 실제로 사용할 제어 문자열이다. 각각 기존의 `\cdot`과 `\textperiodcentered`의 정의를 복사하면 된다.

```
21 \NewCommandCopy{\math@cdot}{\cdot}
```

```
22 \NewCommandCopy{\text@cdot}{\textperiodcentered}
```

두 제어 문자열을 정의했다면 `\cdot`을 재정의한 뒤 스위치를 종료한다.

```
23 \protected\def\cdot{\ifmmode\math@cdot\else\text@cdot\fi}
```

```
24 \fi
```

### 6.4.2 줄임표

`\cdots` 텍스트 모드에서는 줄임표를, 수식 모드에서는 점이 가운데에 찍히는 줄임표를 조판하도록 `\cdots`를 재정의한다. 이 작업은 `\ifkopunct@nocdots` 스위치가 `true`일 때에만 진행된다.

```
25 \ifkopunct@nocdots\else
```

`\math@cdots` `\cdot`과 마찬가지로, `\cdots`를 정의할 때에도 `\math@cdots`와 `\text@cdots`가 필요하다. `\math@cdots`는 간단히 `\cdots`의 기존 정의를 복사하면 된다.

```
26 \NewCommandCopy{\math@cdots}{\cdots}
```

`\text@cdots` `\text@cdots`는 엔진에 따라 정의 방법이 다르다. 먼저 유니코드 엔진에서는 간단히 줄임표 문자( $\cdots$ , U+2026)를 `\text@cdots`로 정의하면 된다.

```

27 \iftutex
28 \def\text@cdots{...}

```

레거시 엔진에서는 `\unichar`과 `\DeclareUnicodeCharacter`를 사용하여 다음과 같이 정의한다. 줄임표 문자를 소스 파일에 직접 입력하더라도 줄임표가 올바른 모양으로 조판된다.

```

29 \else
30 \DeclareTextCommandDefault{\text@cdots}{\unichar{"2026}}
31 \DeclareUnicodeCharacter{2026}{\unichar{"2026}}
32 \fi

```

두 제어 문자열을 정의했다면 `\cdots`를 재정의한다.

```

33 \protected\def\cdots{\ifmmode\math@cdots\else\text@cdots\fi}

```

`\Cdots` 6점 줄임표는 `\Cdots`로 정의한다.

```

34 \protected\def\Cdots{%
35 \ifmmode\math@cdots\math@cdots\else\text@cdots\text@cdots\fi
36 }

```

`\hellipsis` `\hellipsis`가 정의되어 있지 않다면 이를 정의한다. 이는 `XYTeX-ko` 및 `LuaTeX-ko`와의 호환을 위한 것이다.

```

37 \@ifundefined{hellipsis}{%
38 \def\hellipsis{.....}%
39 }{}

```

`\textellipsis` 유니코드 엔진에서도 `\dots`나 `\ldots`, `\textellipsis`가 레거시 엔진의 줄임표와 동일한 기호를 조판하도록 재정의한다.

```

40 \iftutex
41 \UndeclareTextCommand{\textellipsis}{TU}
42 \fi

```

줄임표와 관련된 모든 매크로 정의가 완료되었으면 스위치를 종료한다.

```

43 \fi

```

### 6.4.3 낫표와 화살괄호

`\lnat` 왼쪽 홑낫표는 `\lnat`으로, 오른쪽 홑낫표는 `\rnat`으로 조판한다.

```

\rnat 44 \newcommand*\lnat{「}\newcommand*\rnat{」}%

```

`\nat` 이를 사용하여 인자를 홑낫표로 감싸는 `\nat`을 정의한다.

```

45 \newcommand*\nat[1]{\{\lnat{#1}\rnat}\}%

```



`\lNat` 겹낫표를 조판하는 `\lNat`, `\rNat`, `\Nat`도 동일한 방법으로 정의한다.

```
\rNat 46 \newcommand*\lNat{『}\newcommand*\rNat{』}%
\Nat 47 \newcommand*\Nat[1]{\{\lNat{#1}\rNat\}}%
```

`\lhwa` 홀화살괄호는 `\lhwa`, `\rhwa`, `\hwa`로 입력한다.

```
\rhwa 48 \newcommand*\lhwa{<}\newcommand*\rhwa{>}%
\hwa 49 \newcommand*\hwa[1]{\{\lhwa{#1}\rhwa\}}%
```

`\lHwa` 겹화살괄호는 `\lHwa`, `\rHwa`, `\Hwa`로 입력한다.

```
\rHwa 50 \newcommand*\lHwa{<>}\newcommand*\rHwa{>}%
\Hwa 51 \newcommand*\Hwa[1]{\{\lHwa{#1}\rHwa\}}%
```

매크로 정의가 완료되었으면 스위치를 종료한다.

```
52 \fi
```

#### 6.4.4 물결표 재정의

레거시 엔진에서는 물결표를 더 보기 좋게 조판하기 위해 `\textasciitilde`를 재정의한다. 재정의 작업은 레거시 엔진에서 `\ifkopunct@notilde` 스위치가 `false`일 때에만 진행된다.

```
53 \iftutex\else
54 \ifkopunct@notilde\else
```

`\kopunct@tilde` `\kopunct@tilde`는 올바른 형태의 물결표를 조판하는 제어 문자열이다.

`\kopunct@f@size` `\kopunct@tilde`의 정의에는 현재 글꼴 크기를 저장하는 변수 `\kopunct@f@size`와 `\kopunct@f@baselineskip` 줄 높이를 저장하는 변수 `\kopunct@f@baselineskip`이 필요하다.

```
55 \newlength\kopunct@f@size
56 \newlength\kopunct@f@baselineskip
```

`\TildeScaleFactor` 물결표의 모양을 조정하는 변수 세 개를 정의한다.

```
\TildeRaise 57 \newcommand\TildeScaleFactor{1.5}%
\TildeHeight 58 \newcommand\TildeRaise{-1.35ex}%
59 \newcommand\TildeHeight{1ex}%
```

이상의 변수들을 바탕으로 `\kopunct@tilde`를 정의한다. 글꼴 크기를 `\TildeScaleFactor` 배 키운 뒤, 이를 `\TildeRaise`에 저장된 값만큼 상하로 이동하여 틸데 악센트를 조판한다.

```
60 \def\kopunct@tilde{%
61 \setlength\kopunct@f@size{\f@size pt}%
62 \setlength\kopunct@f@baselineskip\f@baselineskip
63 \raisebox{\TildeRaise}{\TildeHeight}[Opt]{%
```

```

64      \fontsize{\TildeScaleFactor\kopunct@f@size}%
65      {\baselinestretch\kopunct@f@baselineskip}%
66      \selectfont\char"7E
67  }%
68  }%

```

\~ 이제 \~{}이 \kopunct@tilde를 사용하도록 \~를 재정의한다.

```

69  \DeclareTextCompositeCommand{\~}{OT1}{\kopunct@tilde}
70  \DeclareTextCompositeCommand{\~}{T1}{\kopunct@tilde}

```

매크로 재정의가 완료되었으면 스위치를 종료한다.

```

71  \fi
72  \fi

```

## 6.5 oblivoir 호환 매크로 정의

oblivoir 호환 매크로는 \ifkopunct@oblivoir 스위치가 true일 때에만 정의된다.

```

73 \ifkopunct@oblivoir

```

### 6.5.1 가운데점

\cntrdot 가운데점은 \cntrdot으로 조판한다.

```

74  \NewCommandCopy{\cntrdot}{\textperiodcentered}

```

\cntrdot은 kopunct의 \cdot과는 달리 수식 모드에서는 사용할 수 없다.

### 6.5.2 줄임표

\cntrdots 줄임표는 \cntrdots로 조판한다. 한국어에서 사용하는 형태의 줄임표가 조판된다.

줄임표 문자를 직접 입력하더라도 동일한 결과를 얻는다.

```

75  \iftutex
76  \protected\def\cntrdots{...}
77  \else
78  \protected\def\cntrdots{\unichar{"2026}}
79  \DeclareUnicodeCharacter{2026}{\unichar{"2026}}
80  \fi

```

\obldots 영어의 줄임표는 \obldots로 조판한다.

```

81  \def\obldots{%
82    .\kern\fontdimen3\font
83    .\kern\fontdimen3\font
84    .\kern\fontdimen3\font
85  }

```

`\obellipsis` 한국어의 6점 줄임표는 `\obellipsis`로 조판한다. `oblivoir` 클래스의 정의를 그대로 사용하였다.<sup>3</sup>

```
86 \protected\def\obellipsis{\textellipsis\textellipsis}
```

이상의 매크로는 `kopunct`의 `\cdots`와 달리 수식 모드에서는 사용할 수 없다.

### 6.5.3 낫표와 화살괄호

`\snm` `\snm`은 인자를 하나 입력받아 홑낫표로 감싸 준다.

```
87 \def\snm#1{「#1」}
```

`\bnm` `\bnm`은 인자를 하나 입력받아 겹낫표로 감싸 준다.

```
88 \def\bnm#1{『#1』}
```

`\cnm` `\cnm`은 인자를 하나 입력받아 홑화살괄호로 감싸 준다.

```
89 \def\cnm#1{<#1>}
```

`\ccnm` `\ccnm`은 인자를 하나 입력받아 겹화살괄호로 감싸 준다.

```
90 \def\ccnm#1{《#1》}
```

이상의 매크로는 `\nat`, `\hwa`와 달리 인자들을 별도의 영역(group)으로 묶은 뒤 조판하지 않는다. 따라서 인자에 선언형 명령어를 사용할 때 주의해야 한다.

### 6.5.4 줄표

`\expldash` `\expldash`는 줄표를 조판한다. `oblivoir`와는 다르게 그냥  $\TeX$ 의 엠 대시를 사용한다.

```
91 \protected\def\expldash{\,---\,}
```

`\explpunc` `\explpunc`도 정의되어 있다. `oblivoir`의 명령어와 동일한 방법으로 사용한다.

```
92 \protected\def\explpunc.#1.\ {\,---#1---\,}
```

매크로 정의가 완료되었으면 스위치를 종료한다.

```
93 \fi
```

## 찾아보기

이탤릭체로 적힌 숫자는 해당 항목에 대한 설명이 있는 페이지를 가리키며, 밑줄 쳐진 숫자는 해당 항목을 정의하는 코드의 줄 번호를 가리킨다. 정체로 쓰인 숫자는 항목이 사용된 코드의 줄 번호를 가리킨다.

---

<sup>3</sup>따라서 `oblivoir` 옵션을 사용한 경우에는 모든 엔진에서 한국어 줄임표가 올바르게 조판되지만, `oblivoironly` 옵션을 사용한 경우에는 유니코드 엔진에서만 한국어 줄임표가 올바르게 조판된다. 레거시 엔진에서는 ‘.....’로 나타난다.

\,,	91,	92	\explpunc,	92	\nHwa,	3
\_,		92	\fi,	23, 24, 32, 33,	\nNat,	3
\~,	4,	69		35, 42, 43,	\obellipsis,	86
\baselinestretch,		65		52, 71, 72, 80, 93	\obl dots,	81
\bnm,		88	\font,	82, 83, 84	\ProcessOptions,	17
			\fontdimen,	82, 83, 84	\protected,	23, 33, 34,
\ccnm,		90	\fontsize,	64		76, 78, 86, 91, 92
\cdot,	2,	20	\hellipsis,	3, 37	\ProvidesPackage,	2
\Cdots,	2,	34	\Hwa,	3, 50		
\cdots,	2,	25	\hwa,	3, 48	\raisebox,	63
\char,		66			\relax,	17
\cnm,		89	\ifmmode,	23, 33, 35	\RequirePackage,	18
\cntrdot,		74	\iftutex,	27, 40, 53, 75	\rHwa,	50
\cntrdots,		75			\rhwa,	3, 48
			\kern,	82, 83, 84	\rNat,	46
\DeclareOption,	9,		\ldots,	3	\rnat,	3, 44
	10, 11, 12, 13		\lHwa,	3, 50	\selectfont,	66
\DeclareTextCommandDefault,	30		\lhwa,	3, 48	\setlength,	61, 62
\DeclareTextCompositeCommand,			\lNat,	3, 46	\snm,	87
	69, 70		\lnat,	3, 44		
\DeclareUnicodeCharacter,			\Nat,	3, 46	\textellipsis,	40, 86
	31, 79		\nat,	3, 44	\textperiodcentered,	
\def,	23, 28, 33, 34,		\NeedsTeXFormat,	1		22, 74
	38, 60, 76,		\newcommand,	44, 45,	\TildeHeight,	4, 57, 63
	78, 81, 86, 87,			46, 47, 48, 49,	\TildeRaise,	4, 57, 63
	88, 89, 90, 91, 92			50, 51, 57, 58, 59	\TildeScaleFactor,	
\dots,	3		\NewCommandCopy,			4, 57, 64
					\UndeclareTextCommand,	
\else,	19, 20, 23, 25, 29,			21, 22, 26, 74		41
	33, 35, 53, 54, 77		\newif,	4, 5, 6, 7, 8		
\expldash,	91		\newlength,	55, 56	\unichar,	30, 31, 78, 79